

MECHANISMS AND FORMAL DESCRIPTIONS FOR TRADING SERVICE OBJECTS IN GRID MARKETS

Björn Schnizler, Benjamin Blau

Information Management and Systems, University of Karlsruhe (TH), Germany

E-mail: schnizler|blau@iism.uni-karlsruhe.de

Abstract

Along with the growth of Grid infrastructures over the last years, ubiquitous service provision and consumption has received wide attention in literature and practice. Although the number of available services rises rapidly, concepts and standards for an economical efficient service allocation remain in their infancy. Most proposed mechanisms lack of an interdisciplinary analysis that accounts for economical and technical requirements of the underlying trading objects.

In this paper we attempt to reduce this gap by introducing concepts and prescriptive languages for describing services as trading objects. Our contribution is a thorough classification of potential trading objects in Grid markets, an analysis of formal languages to describe these objects, and an evaluation of potential trading mechanisms.

1 Introduction

The Grid is a promising technology for providing access to distributed high-end computational capabilities. Computational tasks can be performed ad-hoc by other resources in the Grid that are not under the user's control. The specification of open standards that define the interactions between different computing resources across organizational entities represent a significant milestone in this development. With the Open Grid Service Architecture (OGSA), the Grid Community has laid the foundation for future developments. OGSA defines computer and storage resources as well as networks, applications, databases and the like as services, i.e. network-enabled entities that provide certain capabilities. Thinking of resources as services paves the way for interoperability among heterogeneous computing and application environments and results in service-oriented systems.

The implementation of Grids has major ramifications, since organizations that have computational demand are not

required to purchase and maintain computer resources on their own. Instead, it is possible that computation can be performed on demand by using resources that are not under permanent control of the temporary user. The demand for resources is covered by service owners that have temporarily idle computers. They are either small-scale owners (such as private persons) or large-scale owners (such as computer center operators).

A lot of research has been devoted to the development of Grid middleware [10]. Current middleware provides the technical infrastructure that allows the sharing of services over multiple geographic and administrative domains. However, the allocation of the supplied services to jobs has been studied in less detail. Current resource management systems typically implement idiosyncratic cost functions for scheduling jobs. Those mechanisms, however, are central in nature. They work well if information about supply and demand is reported truthfully. However, since Grid technology addresses resources sharing cross organizational frontiers, centrally controlled mechanisms suffer untruthful revelation of job related data.

Market mechanisms are known to attain fairly efficient allocations in situations where the participating agents may reveal their private information about costs and valuations. In this context, several auction mechanisms and bargaining protocols for the Grid were proposed to allocate resources efficiently [6, 22]. However, none of the proposed mechanisms have yet made it into practice [15]. One reason for this lies in the abstraction of the underlying trading objects. When referring to markets in the Grid, it is essential to consider the technical fundamentals and to understand what can really be traded and how. This requires a full understanding of the trading object and its characteristics. Oftentimes, this understanding is not fully integrated into the proposed mechanisms.

The objective of this paper is to reduce this gap between market mechanisms and technical specifications in service-oriented systems. We provide a thorough classification for potential trading objects and suitable formal descriptions in the Grid with regard to technical feasibilities. Our contri-

bution is twofold: On the one hand, we provide a detailed technical and economical analysis and classification of different service types and suitable description languages for characterizing their properties. On the other hand, we give recommendations of potential market mechanisms for trading different types of services. Our goal is to provide a broadly diversified view from a technical and an economical perspective on Grid services and to combine these aspects into an overall trading concept.

The paper is structured as follows: In Section 2, we introduce a classification schema for potential trading objects in a service market. Based upon this schema, Section 3 introduces a set of formalisms to describe these trading objects in order to interchange their properties automatically. In Section 4, we review existing market mechanisms with regard to their applicability for trading different types of service. Finally, Section 5 concludes the paper.

2 Classification of Trading Objects

Trading objects in a Grid market means trading rights to use certain computational resources on different machines [23]. Such resources are heterogeneous in regard to their capabilities and their potential fields of application. Some of them may attract various agents while others are irrelevant for most users. Accordingly, the number of potential market participants depends on the type of resource being traded. As the number of market participants has a profound impact on the design of a market mechanism, different types of resources are analyzed with respect to their expected degree of supply and demand.

Figure 1 depicts a three layered view on potential services in the Grid: *Standardized elementary services*, *standardized application services*, and *non-standardized application services* construct each layer respectively.¹ Furthermore, the bottom line represents physical resources through which the service is provided. These resources can be either primitive resources (e.g., a CPU or a sensor) or a set of resources in the form of a cluster. Physical resources are, however, not potential trading objects in the Grid, as they are virtualized by the middleware.

2.1 Standardized Elementary Services

The first layer comprises services that virtualize physical resources. This includes, for instance, a computational service that virtualizes a cluster or a desktop machine. These resources are denoted as standardized elementary services. Although type and behavior of these services are mostly standardized, the services have multiple attributes

¹Services can also be classified by a more granular distinction. However, for the work at hand, the proposed three-layered view suffices.

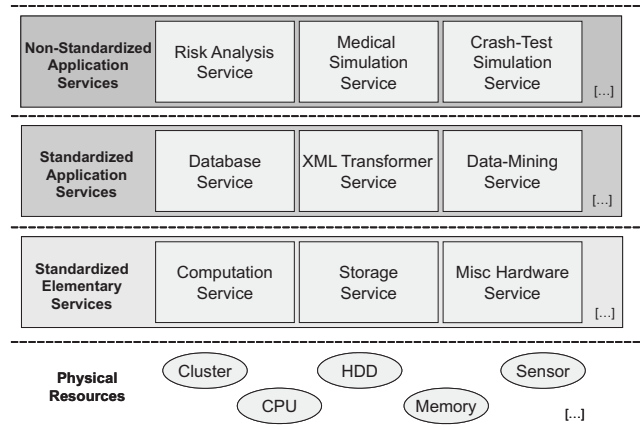


Figure 1. Layered view on different service types in the Grid

with varying characteristics. For instance, storage services may differ according to their capacity (in Gigabyte GB), access time (in milliseconds (ms)), and data throughput (in bits per second (bits/s)). These varying characteristics of the same type of resource, as well as the resource itself can be described by means of standardized description languages such as RSL [8] or GLUE [1]. As these services virtualize physical resources, they constitute elementary entities for a Grid that are required by various different users and applications.

2.2 Standardized Application Services

Application services with a broad scope of use are represented by the second layer. The input and output semantics of these so-called standardized application services are well-accepted and interpretable by a major part of Grid users. Exemplars might be database services and XML transformation services. Services in this layer are required for several different higher-level applications and, as a consequence, are utilized by a multitude of different users. They may also utilize services of the layer below. For instance, a database service makes use of a storage service. Similar to elementary services, the provided quality of service levels for the same type of service may vary. For instance, a set of XML transformation services may vary from their offered response time (in ms); however, it is assumed that these characteristics can also be described in a standardized form.

2.3 Non-Standardized Application Services

Services represented by the third layer are non-standardized application services. Such services are only used for specific application areas such as a simulation service that is required for medical research. Hence, these services are oftentimes only utilized by a small number of Grid participants. Usually, their characteristics and capabilities cannot be described by means of standardized, well-accepted, and interpretable description languages. In addition, the meaning of the services and their attributes are usually vague and can only be interpreted by experts.

3 Formal Descriptions for Trading Objects

Trading services in a Grid by means of a market mechanism requires a precise specification of the object's characteristics and properties in a standardized manner. In order to describe and exchange information on various trading objects, suitable formalisms have to be evaluated with respect to the type. While standardized service objects can be described by well-established and widely-used formalism (e.g. WSDL, WS-BPEL), non-standardized service objects are too complex to be characterized by a single set of attributes. Instead we propose that a trading process of non-standardized service objects should be supported by rich underlying semantic information such as ontologies, taxonomies and constraints. Semantically enhanced formalisms may be useful to enrich description expressiveness to support efficient object allocation among market participants. Furthermore standards such as WS-Agreement support bargaining processes and object matchmaking between a resource provider and a consumer by creating agreement instances out of agreement templates. The description languages proposed in this section are compatible with this standard and can be embedded in WS-Agreement. Especially in collaborative forms of negotiation, it is important to provide a conceptual and shared understanding of externalized information (functional and non-functional) describing properties of trading objects for which semantic approaches are inevitable.

3.1 Standardized Elementary Services

Standardized elementary services build an encapsulating layer for accessing and managing physical resources. Hence, elementary services and the resources they manage form a service-resource-bundle. Therefore it is important not only to describe services on top of the resource layer but also attributes and properties of the underlying resources themselves. This section illustrates state of the art description languages that standardize physical resources

(e.g. storage, database) representing properties and state information for elementary services on top.

Resource Specification Language (RSL) The Resource Specification Language [8] addresses the Globus Job Management System and defines a specification concept for describing resource needs of submitted jobs. Attribute-Value-Pairs describe and control the behavior of one or more components in the resource management system. RSL is based on a simple string-based description syntax which is not based on any existing standard.

Grid Laboratory for a Uniform Environment (GLUE)

GLUE schema [1] provides a description for resources in Grid Information Services (GIS) which defines an information model that is an abstract of the real world with focus on IT-infrastructure elements (e.g. servers, computers, databases). The GLUE schema is a subset of the Common Information Model (CIM) [9]. The core entities within a GLUE schema include a site concept, an abstraction of a service concept as well as computing and storage elements. In comparison to RSL, GLUE is based on an object-oriented concept which provides a higher expressiveness including simple semantic information such as hierarchy and taxonomies.

Service Modeling Language (SML) With the Service Modeling Language [2] the SML consortium² tries to establish a standard for modeling and exchanging system and resource models. Entity information covers aspects like configuration, deployment, monitoring, policy, capacity planning, target operating range and service level agreements.

SML consists of a profile of XML Schema and Schematron and extensions for typed inter-document references (see Listing 1). Based on these construct, it is possible to validate the correctness of models.

Additionally the Service Modeling Language Interchange Format (SML-IF) is a standardized exchange document which encapsulates models, model instances and rules.

In some cases, pure syntactic information on resource properties might not be sufficient for exposing detailed resource information with regard to an efficient trading process. SML provides a flexible basis for annotating entities and relationships with concepts from an external semantic domain for exposing additional information on services and resources.

Listing 1 represents a model for an *ApacheServer* which is described by attributes like *ID* and *version*

²Consists of members from BEA, BMC, CA, Cisco, Dell, EMC, HP, IBM, Intel, Microsoft, and Sun

and implements one or more *ApacheModules*. This excerpt shows one of the SML extensions to XML schema which allows inter-document references.

```

1 <xs:element name="ApacheModule">
2   <xs:complexType>
3     <xs:sequence>
4       <xs:element name="Type" type="xs:string"/>
5       <xs:element name="Version" type="xs:string"/>
6       <xs:any namespace="##any"/>
7     </xs:sequence>
8     <xs:anyAttribute namespace="##any"/>
9   </xs:complexType>
10 </xs:element>
11 <xs:element name="ApacheServer">
12   <xs:complexType>
13     <xs:sequence>
14       <xs:element name="ID" type="xs:string"/>
15       <xs:element name="Version" type="xs:string"/>
16       <xs:element name="ApacheModules" minOccurs="0">
17         <xs:complexType>
18           <xs:sequence>
19             <xs:element ref="tns:ApacheModule"/>
20           </xs:sequence>
21         </xs:complexType>
22       </xs:element>
23     </xs:sequence>
24   </xs:complexType>
25 </xs:element>

```

Listing 1. SML Server Model

3.2 Standardized Application Services

Standardized application services provide a more complex functionality compared to elementary services. They often utilize compositions of services from the underlying layer in order to fulfill more sophisticated tasks. Nevertheless their attributes can be described in a standardized form. The following paragraph outlines means for describing standardized application services in a syntactic and semantic manner.

3.2.1 Syntactic Description Languages

Syntactic description languages focus on functional properties of service objects (e.g. interfaces, operations). They provide widely-used standards to characterize basic service functionality.

Web Service Description Language (WSDL): The Web Service Description Language (WSDL) [7] can be used to describe network services. It is an abstract definition of an end-point-based communication transporting messages that contain document or procedure information. WSDL separates message descriptions from the actual transported data as well as port types which are abstract definitions of operations being performed.

Business Process Execution Language (WS-BPEL)

WS-BPEL [13] is a standardized description lan-

guage for composing Web services from a business perspective in order to fulfill certain tasks. It is a prescriptive formalism for modeling Web service orchestrations that focus on the business process view of one participant and not on a global scope. The WS-BPEL concentrates on the execution layer of statically bound web services [26] and is based on WSDL.

3.2.2 Semantic Description Languages

Additionally to the previously outlined syntactic description languages this paragraph gives a thorough overview over semantic enhanced standards to characterize service objects and highlights their relevance for service object trading.

Web Service Description Language Semantics (WSDL-S)

WSDL-S defines extensions for WSDL that form semantic annotations. These semantic annotations are realized by model reference attributes that hold URIs as references to the concepts within external domain models. In this way, the elements of WSDL-S documents are annotated with meaning [24].

WSDL-S introduces the extensibility elements *pre-condition* and *effect*. They describe the semantics of operations through references to domain model concepts or through literals in any knowledge representation language. In this way, clients become semantically aware of the operations which are available for influencing their environment. WSDL-S describes a general approach about how to enhance existing syntactical XML descriptions with semantic information. Important is the idea of the external domain model and the agnostic about its representation format, which is achieved through references. OWL ontologies as domain models and semantic annotations within the resource properties documents or their XSDs would be a possible transformation of this approach in the WSRF context.

Web Ontology Language Semantics (OWL-S)

OWL-S [16] supplies Web service providers with a core set of markup language constructs for describing the properties and capabilities of their Web services in unambiguous and computer-interpretable form. OWL-S markup of Web services facilitates the automation of Web service tasks including automated Web service discovery, execution, interoperation, composition and execution monitoring. Following the layered approach to markup language development, the current version of OWL-S builds on top of OWL. OWL-S aims on extending best-practice standard like WSDL/SOAP, UDDI and BPEL but partly overlaps and describes

redundant information. The OWL-S Upper Ontology consists of three components:

Service Profile: The service profile presents what the service provides. It is used to advertise the Web service capabilities like non-functional properties, QoS, descriptions and classifications. The service profile consists of a set of preconditions, input and output specifications, effects, the type of service, a product associated with the service and additional service parameters

Service Model: The service model describes how a service works, e.g. the internal process of the service. It facilitates the Web service invocation, composition and monitoring of Web service interactions. The process definition via a service model consists of input parameters, preconditions and results

Service Grounding: It provides a specification of service access information based on WSDL. In combination with the service model the service grounding provides all the information necessary for using the particular service.

3.3 Non-standardized Application Services

As discussed in Chapter 2 non-standardized application services are characterized by a high complexity and a vast usage of different resources. As a consequence their characteristics and capabilities cannot be described by means of standardized, well-accepted, and interpretable syntactic description languages.

When dealing with complex multi-attributive services it is fundamental to define a common language understanding for description and negotiation. Efficient knowledge exchange about a service is the basis for allocating diverse and heterogeneous trading objects. Hence an ontology [11] consisting of a terminology (domain vocabulary), fundamental concepts, their relationships and a taxonomy has to be defined in order to support the bargaining process.

4 Potential Market Mechanisms

Having classified different service objects, this section reviews different market mechanisms that can be applied to trade services efficiently. Due to the similar conditions of the layers comprising standardized elementary services and standardized application services, they are summarized by one layer for standardized services.

Markets as coordination mechanisms for using Grid services can be established for both proposed types of services.

However, the requirements upon each underlying market mechanism differ due to the different characteristics of the transaction objects. Table 1 illustrates these differences: For standardized services, on the one hand, the number of potential providers and consumers is assumed to be high. The target resources are standardized transaction objects and, in some cases, even standard commodities. The number of different types of standardized services is limited, as only few services exist that are generic and fundamental enough to be utilized in various different fields of application. Due to their high degree of standardization, the provision and utilization of these services can both be automated. Non-standardized services of the same type are, on the other hand, scarce. For instance, only few providers can offer highly specialized medical simulation services of the same type with the same functionality. As a result, the number of potential providers and consumers for such services is assumed to be low. In contrast to standardized services, various different types of non-standardized services exist; however, due to their specificity and their lack of standardized descriptiveness, they can hardly be discovered and invoked automatically.

	Market for trading	
	Standardized Services	Non-Standardized Services
Number of providers and requesters	high	low
Commodity type	standardized	unstandardized and scarce
Different types of resources	low	high
Degree of automated usability	high	low

Table 1. Characterization of standardized and non-standardized services

The different characteristics of standardized and non-standardized services result in diametrical requirements upon a market for trading them. With respect to these differences, experiences gained from traditional procurement scenarios such as reported in [3] can be transferred: Standardized services in the Grid can be compared to manufactured goods in procurement, such as rubbers and DVD players. For such commodities, literature emphasizes the benefits of auctions as adequate transaction mechanisms [17]. Auctions, however, may not be appropriate for trading non-standardized services. These services, just as airplanes or buildings in procurement scenarios, are characterized by their complexity and their individual natures. A consumer of a non-standardized service may require several interaction steps with a supplier in order to clarify con-

figurations and properties of the traded service. As such, communication and coordination interactions between the service counterparts are important requirements for trading non-standardized services. Such interaction capabilities are, however, not given by traditional auctions. For instance, in a sealed-bid auction, bids are the only messages exchanged by the market participants. In such cases, the use of bilateral negotiations may be superior to auctions, as negotiations facilitate communication and coordination among agents.

A holistic market mechanism that meets the requirements of both types of services may not exist. As a result, the different characteristics of potential markets for trading standardized and non-standardized services require individual and contrarily market mechanisms. In the following paragraphs, we analyze different market mechanisms that can be applied to trade the different service types.

4.1 Trading Standardized Services

As stated above, the use of auction mechanisms is deemed promising for trading standardized services. In the following, we briefly review traditional and multi-dimensional auctions with respect to their applicability for trading standardized services.

Traditional auctions such as English auctions or continuous double auctions can be effectively implemented for trading commodity services. In this context, Sutherland was one of the first who proposed the use of traditional auction mechanisms for allocating homogenous computer resources [25]. He applies a modified English auction for the allocation of a PDP-1³ at Harvard University. In experimental studies, the author shows that the use of auctions increases the total utilization of the PDP-1 significantly. The proposed approach points the way towards market-based allocation of computer resources.

The Popcorn system [19] extends this concept to Internet based systems. Popcorn implements three different kinds of traditional auction mechanisms for the allocation of distributed computation cycles: It applies Vickrey auctions, first-price double auctions, and a k -price double auctions. The system is evaluated through field experiments and simulations. As a result of their experiments, the authors show that all applied mechanisms achieve approximately efficient outcomes.

Traditional auction mechanisms can be effectively applied to allocate homogenous computer resources. However, the mechanisms cannot be directly applied to modern service oriented systems as they do not address service specific requirements. They neither support the allocation of service bundles (e.g., a bundle that consists of a storage service and a database service) nor support services with mul-

multiple attributes (e.g., a storage service that can be characterized by its size and its speed). For such services, the application of more complex auction mechanisms is superior. Exemplarily for service bundles we presented WS-BPEL as a description language for modeling service composition. Depending on the preferences of the service consumer a composition of services might yield a higher overall utility than the sum of the values of the same unbundled services. In this case, the services should be allocated by means of *combinatorial auction* to result in economically efficient allocations [21]. Combinatorial auctions are multi-item auctions, where an agent can submit bids on multiple heterogeneous services as a bundle. As such, agents can express super-additive utility functions by means of expressing valuations for a bundle of services. A bundle consists of logical AND concatenated bids on a set of services. Such bids ensure that an agent is allocated to either all services of the bundle or to none of it.

In service oriented systems, the efficient application of combinatorial mechanisms has already been evinced [4, 22, 12]. Extending the concepts of combinatorial mechanisms, Schnizler et al. propose a multi-attribute combinatorial exchange called MACE [22]. The auction is a combinatorial mechanism which supports bids on bundles, quality, and time attributes. With regard to the analysis of the authors, MACE qualifies for trading bundles of standardized services that have multiple attributes and timing constraints.

4.2 Trading Non-Standardized Services

Non-standardized services cannot be formalized using standardized description languages. Thus, traditional auction formats are hardly applicable to this type of services. Traditional auction formats require simple types of trading objects mainly be described by a price. For these auctions complex ontologies, taxonomies and protocols are not necessary [5]. We think that the process of allocating non-standardized services involves negotiation protocols respectively bargaining mechanisms. For instance, Grid related systems such as CATNETS [20], OCEAN [18] and Nimrod/G [6] implement a variety of bargaining schemas. In the future, these mechanisms may be supported by the dedicated negotiation support systems [14] or may be automated completely by means of software agents.

5 Conclusion

In this paper we gave a thorough classification of different types of service objects in Grids. According to this classification, suitable description languages were evaluated with respect to trading and bargaining processes in Grid markets. Potential market mechanisms were illustrated and

³The PDP-1 (Programmed Data Processor-1) is a computer system manufactured in 1960.

analyzed with focus on efficient allocation of service objects between market participants. We state that a technical and economical view on services and market mechanisms is necessary to understand how a basis for trading can be established and how the trading process itself can be designed efficiently.

Based on the ideas of this paper, market mechanisms for different types of trading objects will be implemented in the *meet2trade*⁴ [27] market engineering platform and evaluated based on their allocation efficiency.

References

- [1] S. Andreozzi, S. Burke, L. Field, S. Fisher, B. Konya, M. Mambelli, J. M. Schopf, M. Viljoen, and A. Wilson. Glue schema specification version 1.2. Draft specification, 2005.
- [2] J. Arwe, J. Boucher, P. Dubliss, Z. Eckert, D. Ehnebuske, J. Hass, S. Jerman, H. Kreger, V. Kowalski, M. Milenkovic, B. Murray, P. Prasek, J. Saiyed, H. Sluiman, B. Tabbara, V. Tewari, W. Vambenepe, M. Waschke, and A. Westerinen. <http://www.serviceml.org/>, 2007. Service Modeling Language (SML).
- [3] P. Bajari, R. McMillan, and S. Tadelis. Auctions versus negotiations in procurement: An empirical analysis. Working paper, Duke University and Stanford University, 2004.
- [4] R. Bapna, S. Das, R. Garfinkel, and J. Stallaert. A market design for Grid computing. *INFORMS Journal of Computing*, forthcoming, 2007.
- [5] M. Bichler. A roadmap to auction-based negotiation protocols for electronic commerce. *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, page 10, 2000.
- [6] R. Buyya, D. Abramson, and J. Giddy. A case for economy Grid architecture for service oriented Grid computing. In *Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS-01)*, San Francisco, pages 83–98, Washington, DC, USA, 2001. IEEE Computer Society.
- [7] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. <http://www.w3.org/tr/wsdl>, 2001. Web Services Description Language (WSDL) 1.1.
- [8] K. Czajkowski, I. T. Foster, N. T. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke. A resource management architecture for metacomputing systems. In D. G. Feitelson and L. Rudolph, editors, *Proceedings of the Workshop on Job Scheduling Strategies for Parallel Processing (IPPS/SPDP'98)*, volume 1459 of *Lecture Notes in Computer Science*, pages 62–82, London, UK, 1998. Springer.
- [9] DMTF. <http://www.dmtf.org/standards/cim/>. Common Information Model (CIM) Standards.
- [10] I. Foster and C. Kesselman. *The Grid 2 - Blueprint for a New Computing Infrastructure*, volume 2. Elsevier, 2004.
- [11] D. Gaaevic, D. Djuric, V. Devedzic, and B. Selic. *Model Driven Architecture and Ontology Development*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [12] D. Grosu and A. Das. Combinatorial auction-based resource allocation in Grids. *International Journal of Computational Science and Engineering*, forthcoming, 2006.
- [13] D. Jordan and J. Evdemon. <http://www.oasis-open.org/committees/>. OASIS Web Services Business Process Execution Language (WSBPEL) Technical Committee.
- [14] G. E. Kersten. E-negotiation systems: Interaction of people and technologies to resolve conflicts. Presented at: UN-ESCAP Third Annual Forum on Online Dispute Resolution Melbourne, Australia, July 2004, 2004.
- [15] K. Lai. Markets are dead, long live markets. Technical Report 0502027, HP Labs, 2005.
- [16] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, Y. University, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara. <http://www.w3.org/submission/owl-s/>, 2004. OWL-S: Semantic Markup for Web Services.
- [17] P. Milgrom. *Putting Auction Theory to Work*. Cambridge University Press, 2004.
- [18] P. Padala, C. Harrison, N. Pelfort, E. Jansen, M. Frank, and C. Chokkareddy. OCEAN: The open computation exchange and arbitration network, a market approach to meta computing. In *Proceedings of the Second International Symposium on Parallel and Distributed Computing 2003*, pages 185–192. IEEE Computer Society, 2003.
- [19] O. Regev and N. Nisan. The POPCORN market: Online markets for computational resources. *Decision Support Systems*, 28(1-2):177–189, 2000.
- [20] M. Reinicke, W. Streitberger, and T. Eymann. Evaluation of service selection techniques in service oriented computing networks. In *Proceedings of the first workshop on Smart Grid Technologies*, 2005.
- [21] M. Rothkopf, A. Pekec, and R. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.
- [22] B. Schnizler, D. Neumann, D. Veit, and C. Weinhardt. Trading Grid services – a multi-attribute combinatorial approach. *European Journal of Operational Research*, forthcoming, 2006.
- [23] J. Shneidman, C. Ng, D. C. Parkes, A. AuYoung, A. C. Snoreen, A. Vahdat, and B. Chun. Why markets could (but don't currently) solve resource allocation problems in systems. In *Proceedings of Tenth Workshop on Hot Topics in Operating Systems*, 2005.
- [24] K. Sivashanmugam, K. Verma, A. Sheth, and J. Miller. Adding semantics to web services standards, 2003.
- [25] I. E. Sutherland. A futures market in computer time. *Communications of the ACM*, 11(6):449–451, 1968.
- [26] K. Verma, R. Akkiraju, R. Goodwin, P. Doshi, and J. Lee. On Accommodating Inter Service Dependencies in Web Process Flow Composition. *AAAI Spring Symposium PP*, pages 37–43.
- [27] C. Weinhardt, C. van Dinther, K. Kolitz, J. Mäkiö, and I. Weber. *meet2trade: A Generic Electronic Trading Platform*. *4th Workshop on e-Business (WEB 2005)*, Las Vegas, USA, 2005.

⁴meet2trade is a software tool suite designed to systematically support each step of a Market Engineering (ME) process from design to evaluation.